

---

## Tools for Reducing and Managing Link Rot in LibGuides

*While creating content in LibGuides is quite easy, link maintenance is troublesome, and the built-in link checker offers only a partial solution. The authors describe a method of using PURLs and a third-party link checker to effectively manage links within LibGuides.*

By Wilhelmina Randtke and Matthew D. Burrell

---

### Introduction.

This article describes tools for managing links in SpringShare's LibGuides, a popular web platform for libraries. LibGuides includes a built-in link checker for only some links. Instructions are provided on how to run an automated link checker on all links within a single guide. Persistent Uniform Resource Locators (PURLs), hyperlinks which are maintained centrally and rarely break, are recommended for library electronic resources. Special consideration is given to the fact that many libraries using the LibGuides platform may not have easy access to in-house information technology (IT) personnel. A method is presented for implementing PURLs entirely within the LibGuides platform.

---

### The LibGuides Platform.

LibGuides is a popular platform which allows librarians to create simple web pages using standard templates. The LibGuides community directory at <http://libguides.com/community> indicates 2,823 libraries have created 218,355 research guides published in LibGuides as of January 2012. LibGuides is marketed as an online replacement for paper pathfinders, or lists of research sources for a particular topic.[1]

In the summer of 2009, Florida State University (FSU) Libraries purchased LibGuides. Since its implementation, librarians at FSU have used LibGuides extensively and now feature 170 guides available on the libraries' website.

### ***LibGuides allows easy content creation.***

FSU's overall experience with LibGuides has been positive. Librarians appreciate having the ability to manage their own content and be able to simply modify their pages when they see the need. Web Development staff appreciate that an account can be quickly created and managed. LibGuides provides a way for librarians to identify and fix problems with outdated content instantly by editing content in the guide. This eliminates the necessary additional time which may occur on the libraries' websites where only a small group of staff has the ability to update content.

Ease of use and management is significant in light of the fact that many libraries do not have an in-house Information Technology (IT) staff or full time website staff within the library, and may need to rely non-library staff for IT services. While FSU Libraries has a dedicated in-house IT and Web Development team, FSU Libraries recognize that LibGuides is especially valuable for libraries which are not equipped to set up and run a web server.

### ***LibGuides presents universal problems in content maintenance.***

One of the emerging problems as FSU Libraries' LibGuides system matures is broken links, or link rot. As online content relocates and uniform resource locators (URLs) for electronic resources change, links break. Link rot is a universal problem when dealing with online content.

This article seeks to describe ways in which FSU Libraries manage and prevent broken links in the libraries' LibGuides system, and to give step-by-step instructions on link maintenance which can be implemented by libraries either with or without a dedicated IT and web team.

---

### Pitfalls of relying on LibGuides built-in link checker

#### ***Web Links***

LibGuides has a built-in link checker. To get to this a librarian can log into LibGuides, click “My Admin” in the top right corner, and click “Link Checker” at the bottom of the column in the center of the screen.

LibGuides’ link checker has some disadvantages. First, according to the small print explanation at the top of the list of broken links provided by the link checker, “It can only check links that are added within the following box types: Web Links, Links & Lists, Dates & Events, RSS Feeds, Podcasts, Books, User Submits, and Polls. Links entered through the rich text editor are not checked.”[2]

Links in bulleted lists will be checked. However, other links will not. A link inserted in the text of a paragraph cannot be checked using the LibGuides interface. Examination of LibGuides included in the directory at <http://community.libguides.com> demonstrates that it is common for links to be placed in paragraphs of text in the LibGuides system. When links are placed in a paragraph, rather than in a bulleted list, LibGuides provides no link maintenance tools, and libraries need to find other avenues to identify broken links.

### **Electronic Resource links.**

The second weakness of LibGuides’ built-in link checker requires a short explanation of how electronic resources links work. In this article, an electronic resource refers to a database purchased by FSU Libraries. These databases require an institutional subscription, and require authorization recognizing that a patron of FSU Libraries is accessing the resource. Most databases will recognize FSU Libraries patrons based on the IP Address of the computer. If the computer is located on the FSU campus, then the IP address of the computer will show the computer’s location and allow access.

If a patron accesses a database using a computer that isn’t on the campus, then the database needs to have a way of recognizing that the user is a library patron. This is done through a proxy login. A proxy login is a special link, which routes the patron’s computer connection through a server at FSU, and lets the database know that an FSU patron is using the database. For our purposes, a proxy server is seen as a prefix to a URL that shows the IP address of the user as if they were on campus.[3]

Unfortunately, LibGuides’ built-in link checker does not check links with a proxy or any electronic resources which require off-campus login.[4]

### **How to check web links in a LibGuide.**

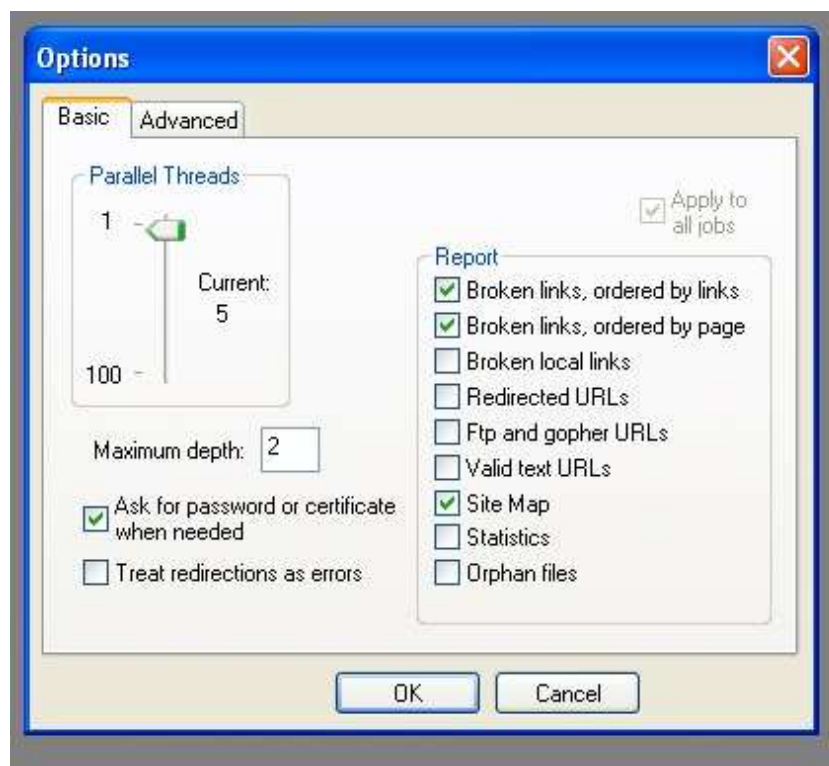
---

Running an external link checker on the LibGuides system is complex. LibGuides uses a “*Robots Exclusion Protocol*” (robots.txt)[5] to instruct search engines not to make cached copies of the guides, and tells the Internet Archive not to spider and store any guides, and which informs some automatic link checkers not to examine the guides.[6] This is why some popular link-checkers, such as the World Wide Web Consortium’s link checker at <http://validator.w3.org/checklink>, cannot be successfully used with LibGuides.

FSU Libraries experimented with link checkers and found that Xenu Link Sleuth will allow a librarian to run an automated link check on his or her guides. Xenu Link Sleuth is a powerful tool for website maintenance, which is available for download at <http://home.snafu.de/tilman/xenulink.html>. In exchange for using the program, the programmer requests donations be made to a list of specified charities, or that a T-shirt, or a thank you letter be sent.[7]

In order to check all outgoing links from a single guide:

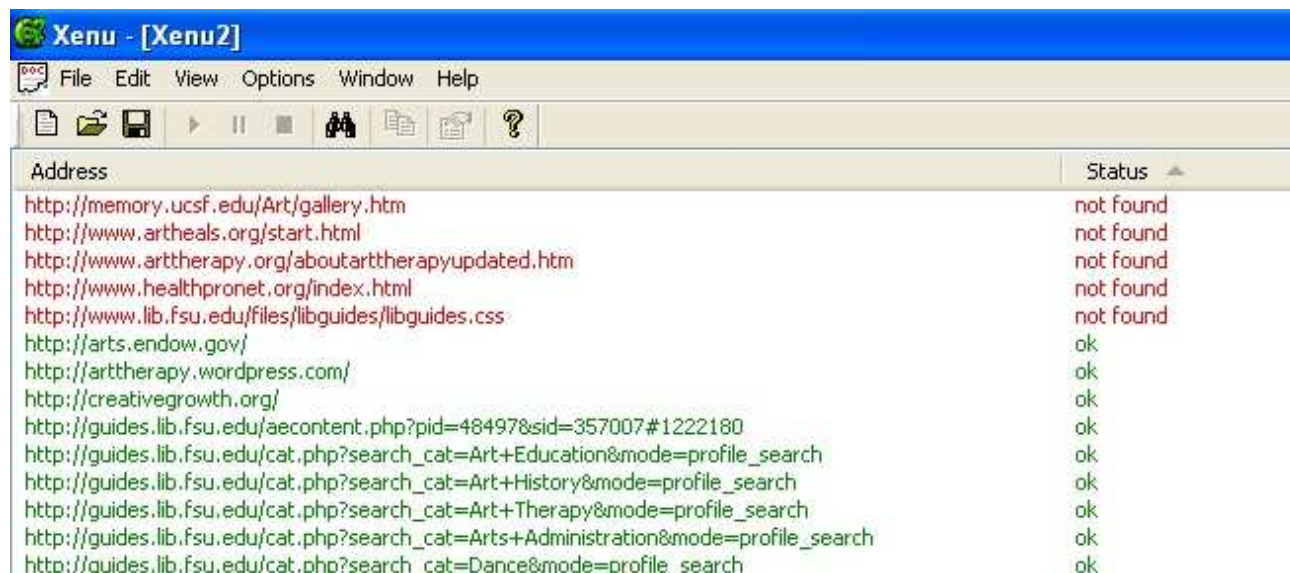
- A. Install and open Xenu Link Sleuth. Go to the <Options> menu and select <Preferences>.
  - B. Set “Maximum depth” to exactly 2. “Maximum depth” determines how many links the link checker follows away from the page you are on. So, 0 means just whether the URL you check loads. 1 means whether links on the page you check load. 2 means whether links on the page you check load, and whether links on those pages load. So, 2 will check the links on the tab of the guide for the URL you input, and for each other tab and subtab in the guide. Putting a higher number would start to check other guides, as well as yours, which will take more time to check and will result in flagging some URLs which you cannot edit because they are not on your guide. (A higher number would be appropriate if you wish to check all LibGuides in the system.)
  - C. Under “Report” make sure “Broken links, ordered by links” or “Broken links, ordered by page” is checked.
  - D. Click <OK>.
  - E. Under the <File> menu select <Check URL...>. (F) Paste in the URL of any one of the tabs in the LibGuide.
  - F. Make sure the “Check external links” is checked. This will check links that go to anywhere outside of your LibGuide (ie. To Google Scholar, a university department’s webpage, etc.)
  - G. Click <OK>.
-



**Figure 1.** In the Preferences menu, Maximum depth should be set to 2 in order to check links within a single guide using Xenu Link Sleuth.

Xenu will check all of the links on a page including images, style sheets, scripting, as well as URL's. Don't be put off by so many links – the routine links will work, and you don't need to worry about links that work.

Once the link checker has finished running, sort the list of links by "Status". Xenu Link Sleuth shows problematic links written in red.



**Figure 2.** Once the link checker finishes, sort links by Status, and right-click on links which are "not found" in order to view the LibGuides tabs within a guide which contain the broken links.

Right click on a broken link and a menu will appear. Scroll down to <URL Properties>. A box will appear and at the bottom of that box will be a list of all the pages that contain the broken link. All the pages should have URLs with the same number after "pid=". (If they don't, then repeat these directions from the top, and especially check that "Maximum depth" was set to exactly 2.) Correction is only necessary for the pages which contain ".../content.php?....". Pages with ".../print\_content.php?...." and ".../content\_mobile.php?..." in the URLs should be ignored, because these are changed automatically when links from ".../content.php?..." pages change.

To correct each link go to the page with a URL containing ".../content.php?..." which the broken link appears on and make the change in LibGuides.

After you have corrected the links, go back to the beginning of this section, and rerun the link checker. Links should be working now.

## **What A link checker can't do.**

---

A word of warning is appropriate. A link checker only checks that when a link is clicked, a page will load. The link checker does not check that the correct page loads. The link checker cannot check most electronic resource links.

The link checker cannot check that a linked page is what a person clicking on the link will expect. For example, if I promise a link to Google Scholar, and then link to Google, to a shopping site, or to any other page on the web, the link checker will find the link, load the page, and report that it found a working page.

Now, picture a link to an electronic resource. Usually, when ProQuest or other vendors change the URLs for databases, the old URLs result not in an HTTP error status, but in a success status, and load a page saying to "Please login" or to pay X amount to view the article or search the database. While a person clicking that link will find it broken, a link checker will see only that a page loaded. A page which requests someone to "Please login" is still a page. The link-checker sees a working link.

No link checker can monitor whether linked electronic resources recognize the user as someone who has or has not legitimately paid for access to the database. And a high proportion of links on a library website will be to electronic resources. Further, some vendors explicitly request that their links not be checked. Therefore, automated link checkers have limited use for libraries.

At this time, the only way to check whether an electronic resource link is working is to click on that link, and retrieve a full text resource from that database. This can be done "on the cheap" by assigning the task of link checking to a worker with less training. It is recommended that a student worker or paraprofessional periodically be assigned to retrieve one full-text resource from each linked resource or database on a library's list of databases. If a full-text resource cannot be retrieved, then the database may not recognize that the user comes from a library which has paid for access. In this case, the webpage and link should be brought to the attention of a librarian who can determine why the connection failure has occurred.

## **Managing Electronic Resource links in LibGuides.**

---

LibGuide has a tool for handling electronic resource links. As described below, this tool does not adequately provide for long-term link maintenance. In 2010, LibGuides partnered with Serial Solutions®. The result of the partnership is that libraries with LibGuides and either a Serial Solutions 360 account or Summon service are able to automatically generate a single LibGuide containing a list of working links to electronic resources.<sup>[8]</sup> This LibGuide can be updated at any later point by running the database import tool. However, this update and A-Z List has some drawbacks. First, if database descriptions are not included in the library's 360 or Summons plan, then in order to include database descriptions, library staff must manually add each description. During subsequent updates, the librarian performing a database update must remember to uncheck the box for "Import database descriptions" while doing the database update. Otherwise, if this box is checked, then a list of empty descriptions will be imported and will overwrite any descriptions librarians have previously created.<sup>[9]</sup> The second drawback is that a database update will not remove databases which the library no longer subscribes to. So, if a subscription is dropped, the link must be removed manually from the A-Z List.<sup>[10]</sup>

The third and most significant drawback is that a list of working links in a single place in the LibGuides system does not mean links from any other guides in the system will work. In LibGuides, a librarian has three options when adding a link. The librarian can create a new link by pasting in a URL, copy a link from elsewhere in the system, or reuse a link. Reusing a link means that when the parent link changes the new link will change as well <sup>[11]</sup>

If all librarians reuse links from the A-Z list and do not copy links, or cut-and-paste the URL, and the administrator for the A-Z List runs regular updates and diligently removes databases, which are no longer subscribed to, then links to electronic resources will be relatively stable.

However, this requires that all librarians using the LibGuides system always "reuse" links from the A-Z List. This is administratively difficult to achieve. In many cases, directions were given months or years before the time when they add or change a link. It is unlikely that all LibGuide account holders will use links to electronic resources from a central list. It is very tempting to cut-and-paste links from elsewhere in LibGuides or from within a database.

## **Persistent URLs: An Old Idea Gets New Life in LibGuides.**

Persistent uniform resource locators (PURLs) refers to a system for creating and maintaining URLs which will always render the same resource even if the resource moves to a new address on the web. This is done by maintaining a list of PURLs and corresponding list of current URLs. FSU Libraries' PURLs may be found at <http://guides.lib.fsu.edu/DatabasesAZ>. The PURLs in this list never change. So the PURLs may be cut-and-pasted, copied, or reused, and may even be added to a

syllabus or course website for use outside of the LibGuides system.

Each PURL on the Databases A-Z LibGuide refers to a database hosted at FSU Libraries which contains the list of PURLs in the A-Z Databases LibGuide and connects that PURL to the correct location of the resource. Proxy logins are provided by the PURL database as well, so librarians do not need to be reminded to include a proxy login in a link to an electronic resource. While FSU Libraries' PURLs use a MySQL database, it is possible to implement PURLs entirely within LibGuides.

### ***How to implement PURLs in LibGuides.***

LibGuides allows librarians and administrators to place a redirect on an entire guide. So, when someone clicks to that guide, the person will automatically be taken to a destination URL specified for that guide, which can be anywhere on the web or in an electronic database. LibGuides also allows one to assign a short human-readable URL to a guide. By combining these two features, it is possible to use LibGuides to create a list of PURLs which is contained entirely within the LibGuides platform.

To create a PURL for a single database: First, log into LibGuides and click "My Admin" in the top right corner. Click the link in the center of the screen to "Create New Guide". For "Guide Title" type the name of your database. For "Redirect URL" type the current URL for the database, including your library's proxy login. Click the button to create this guide.

Now, edit the guide. Along the top of the screen, click the menu for "Guide Setting", and then click "Change Guide Information" in the drop down menu. Change "Guide Publication Status" to Private. That way links to this PURL Guide will work for patrons, but numerous PURL Guides for numerous databases will not clutter the library's LibGuides homepage. Under "Friendly URL" assign the PURL for this database by giving it a short identifiable name. Now click "Save".

Links which point to this guide will always redirect to the correct database URL. If the URL changes, you can simply go to "Change Guide Information", "Advanced Settings" and change the "Redirect URL". Because the "Friendly URL" is something that librarians can recognize as reliable, librarians will constantly use a list of reliable links and know that each is part of the library's PURL system and works.

Create a PURL for each library database. As you go, create your own Databases A-Z guide with the database name, and a link to your PURL.

While PURLs will take a significant amount of time to set up initially, the time will be saved later by consistently linking to the PURLs (guides which redirect to individual databases). So, when a database URL changes, you will only have to correct the link in one single location: the redirect on your PURL guide.

A clear advantage to using PURLs is that links can be inserted in any way by librarians editing the guides. So all three options, cutting and pasting in a new URL, and reusing or copying a URL from elsewhere in LibGuides, will result in a link that is maintained. Another advantage is that by using a consistent naming system for all assigned Friendly URLs, for example, by beginning all friendly URLs with "purl\_\_\_\_" or "data\_\_\_\_" and then an abbreviation for the database, a librarian can quickly mouse over links to view the destination URL, and verify that the destination is a library PURL. So, a librarian can know that the link is reliable without clicking it and retrieving a document from the electronic resource.

PURLs can also be promoted to professors, student organizations, or tutor organizations who can print the PURL in syllabi or handouts and be assured that the link will not break. So, a reliable system of PURLs has potential as a marketing tool for promoting library services outside the library.

### ***Other solutions to identifying broken links***

There are other solutions for identifying broken electronic resource links in LibGuides. One is to ask each librarian to manually check his or her own LibGuides. This takes a considerable amount of time that the librarian may not have, and because the same database may appear on different guides, effort will be duplicated as every occurrence of a link to the database is checked. Further, if the librarian checks links from an office on campus, the librarian will not notice links for which a proxy login should be but was not included. These links will appear to function from an on campus computer, but will not allow a student to retrieve resources from office campus, because the database will not recognize the user as coming from the university which licensed its databases.

### ***Pros and Cons of implementing PURLs in LibGuides.***

---

Implementing PURLs for electronic resource links will likely save staff time in checking for broken links within the system, nevertheless, initially creating the PURLs within LibGuides involves a significant investment of staff time. This investment of time is a concern, because the PURLs within LibGuides cannot be moved to another system. If the library cancels its LibGuides account, then that time is lost, and incoming links in other portions of the library website, and in professors' syllabi will break perhaps causing a lack of confidence in the library. LibGuides may also change its interface in the future, leading to loss of the PURLs. In short, the problem is: Implementing PURLs in LibGuides leads to buy-in.



A more technologically complex solution is to implement PURLs in-house using a database. FSU Libraries has chosen to do this using a MySQL database. MySQL is not necessary, and a comparable system can be implemented in Oracle, or any other database server. Crucially, FSU Libraries has technical staff within the library including multiple staff that can set up a database server and make regular updates to the databases on that server. Continued access and updates are important, because every time an electronic resource link changes, the database must be updated with the new URL. Housing PURLs on a system controlled and owned by the library is recommended for libraries, so long as the library is able to (1) have a database set up by IT staff, (2) be able to have the database updated with a fast turnaround time on a regular basis, and (3) keep the database running, up-to-date, and working consistently indefinitely into the future. Advantages to housing the PURLs in house are significant. It prevents buy-in to a hosted system which is priced on a yearly basis, and may become increasingly expensive over time. It prevents buy-in to a hosted system whose functionalities may be altered or discontinued, without the library's input or consent. And, finally, for staff with technology training sufficient to work with an in-house database, it is possible make regular backups of PURLs and then to restore them later in case of a problem with the database. True system wide back ups are not possible in LibGuides. Libraries with sufficient in-house access to technology and the skills to make regular updates to a database should explore options for housing PURLs in-house. An excellent source to begin exploring in-house options is Tonkin's article Persistent Identifiers: Considering the options, which gives an overview of different ways of hosting PURLs and points to more detailed resources on implementing PURLs using various technologies.[12]

Despite potential buy-in issues with implementing PURLs in LibGuides, the simplicity of the system allows non-technical staff to comfortably and intuitively interact with LibGuides. This prevents bottlenecks which might occur, if technical staff were needed to make simple regular revisions to a database, and technical staff is not easily available. This article has attempted to describe a system of implementing PURLs directed at these libraries with small technical staff or those with limited access to IT. The IT staff and capability of the library to host a database should determine the choice of what tool to use to implement PURLs.

## Conclusions.

---

The LibGuides platform is a useful tool for creating librarian-managed subject or class webpages. As this platform ages the problem of link rot which is universal to online resources is a threat to the accuracy of published library content, as well as the efficacy of a well-maintained library website. The LibGuides platform provides minimal tools for checking and managing links to electronic resources. By using widely available tools any librarian can check web links within his or her own guide. It is also possible to use LibGuides to create and maintain a system of PURLs, or links which refer to a single central location and so will not break as long as that central location is kept current. It is recommended that libraries consider using PURLs to manage electronic resource links.

## References.

---

- Burke, A. 2011a. A-Z Subscription Database List Management [Internet]. [updated 2011 Jul 01]. SpringShare. Accessed at <http://help.springshare.com/azlist>
- Burke, A. 2011b. Box Types [Internet]. [updated 2011 Jul 01]. SpringShare. Accessed at <http://help.springshare.com/boxtypes>
- Hausherr, T. 2011. Find Broken Links on Your Site with Xenu's Link Sleuth [Internet]. [cited 2011 Jul 07]. Available from <http://home.snafu.de/tilman/xenulink.html>
- LibGuides for Libraries [Internet]. [updated 2011]. SpringShare; [cited 2011 Jul 07]. Available from <http://springshare.com/libguides/>
- Link Checker – Subject Guides at The Florida State University [Internet]. [updated 2011 Jul 07]. SpringShare; [cited 2011 Jul 07]. On file with the author.
- Serials Solutions KnowledgeWorks Powers Springshare LibGuides A-to-Z Database List [Datasheet]. [updated 2010]. SerialsSolutions: A ProQuest Company. Accessed at <http://www.serialssolutions.com/assets/resources/Data-Sheet-KnowledgeWorks-Springshare-LibGuides.pdf>

## Notes

---

- [1] LibGuides for Libraries [Internet]. [updated 2012]. SpringShare; [cited 2012 Jan 18]. Available from <http://springshare.com/libguides/>
- [2] <http://guidefaq.com/a.php?qid=4399>
- [3] For more information about proxy servers see the OCLC article at <http://www.oclc.org/us/en/services/brochures>

/213330usf\_ezproxy.pdf.

[4] Email communication: “We filter out (i.e., don’t check) URLs that have multiple “http” strings in them, which means that links using a proxy prefix are not checked.” Sent Jan 18, 2012 from Anna Burke to Matt Burrell

[5] See “View Source on any LibGuide page: ‘robots’ content=’noarchive’

[6] <http://noarchive.net/refs/search/#Google>

[7] <http://home.snafu.de/tilman/xenulink.html#donation>

[8] <http://help.springshare.com/azlist>

[9] Burke, A. 2011a. A-Z Subscription Database List Management [Internet]. [updated 2011 Jul 01]. SpringShare. Accessed at <http://help.springshare.com/azlist>

[10] Ibid.

[11] Ibid.

[12] Tonkin, E. 2008. Persistent identifiers: Considering the options [Internet]. [cited Jan 30, 2012] Ariadne; 56. Available at <http://www.ariadne.ac.uk/issue56/tonkin/>.

## About the Authors

---

Wilhelmina Randtke  
Florida State University Libraries  
Graduate Assistant in Web Development  
Strozier Library, Room 215E  
116 Honors Way  
Tallahassee, FL 32306  
(850) 345-6123  
[wvr05@fsu.edu](mailto:wvr05@fsu.edu)

Matthew D. Burrell  
Florida State University Libraries  
Web Developer  
Strozier Library, Room 209A LIB  
116 Honors Way  
Tallahassee, FL 32306  
(850) 645-6986  
[mdburrell@fsu.edu](mailto:mdburrell@fsu.edu)

Subscribe to comments: [For this article](#) | [For all articles](#)

---

This work is licensed under a Creative Commons Attribution 3.0 United States License.

